CLAIMS

What is claimed is:

1. A processor, comprising:

a Boolean logic unit, wherein the Boolean logic unit is operable for performing the short-circuit evaluation of Conjunctive Normal Form Boolean expressions/operations;

a plurality of input/output interfaces, wherein the plurality of input/output interfaces are operable for receiving a plurality of compiled Boolean expressions/operations and transmitting a plurality of compiled results; and

a plurality of registers.

2. The processor of claim 1, further comprising a plurality of AND gates, wherein the plurality of AND gates are operable for passing a plurality of expression/operation results and signals.

3. The processor of claim 2, wherein the plurality of AND gates comprise an *n*-bit AND register, wherein the *n*-bit AND register is operable for rolling up the results of conjuncts.

4. The processor of claim 3, wherein the default value of the *n*-bit AND register is 'one'.

5. The processor of claim 3, wherein the *n*-bit AND register initializes to a value of 'one' after the start of an operational code.

6. The processor of claim 3, wherein the *n*-bit AND register remains at a value of 'one' if all of the conjuncts of a Boolean expression/operation being evaluated are true.

7. The processor of claim 3, wherein the Conjunctive Normal Form Boolean expression/operation is false if the *n*-bit AND register is set to 'zero', and the remainder of the Boolean expression/operation is short-circuited.

8. The processor of claim 1, further comprising an OR gate, wherein the OR gate is operable for passing a plurality of expression/operation results and signals.

9. The processor of claim 8, wherein OR gate comprises a *n*-bit OR register, wherein the *n*-bit OR register is operable for rolling up the results of conjuncts.

10. The processor of claim 9, wherein the *n*-bit OR register initializes to a value of 'zero' and remains in that state until a state in a predetermined conjunct evaluates to 'one'.

11. The processor of claim 10, further comprising an *n*-bit OR conjunct register, wherein the *n*-bit OR conjunct register indicates that the evaluation of a conjunct comprising an OR clause has begun,

12. The processor of claim 11, wherein the *n*-bit OR conjunct register initializes to a value of 'zero' and remains in that state until an OR expression/operation sets its value to 'one'.

13. The processor of claim 11, wherein a predetermined conjunct evaluates to true if the *n*-bit OR register is set to 'one' and the *n*-bit OR conjunct register is set to 'one', and the processor short-circuits to the start of the next conjunct.

14. The processor of claim 1, further comprising an operation decoder, wherein the operation decoder is operable for deciphering an operational code and controlling units that are dependent upon the operational code.

15. The processor of claim 14, wherein functions of the operation decoder comprise Boolean AND, Boolean OR, end of operation, no operation, unconditional jump, conditional jump, start of operation, and start of conjunct.

16. The processor of claim 1, further comprising a control encoder, wherein the control encoder accepts $n+m$ bits in parallel and outputs them across a device bus either in series or in parallel.

17. The processor of claim 1, further comprising a random-access memory, wherein the random-access memory is operable for storing the states of a plurality of devices that the processor monitors and controls.

18. The processor of claim 1, further comprising a memory, wherein the memory is operable for holding a compiled micro-program.

19. The processor of claim 18, further comprising a program counter, wherein the program counter is operable for fetching an instruction from the read-only memory.

20. The processor of claim 19, further comprising a memory device, wherein the memory device is operable for configuring the program counter for normal operation, unconditional jump operation, conditional jump operation, and Boolean short-circuit operation.

21. The processor of claim 1, wherein the plurality of registers comprise a plurality of multi-bit registers.

22. The processor of claim 21, wherein the plurality of multi-bit registers comprise an instruction register, a next operation address register, and an end of OR address register.

23. The processor of claim 22, wherein the instruction register comprises an $n+m+3$-bit wide register comprising an n-bit address, an m-bit control/state word, and a 3-bit operational code.

24. The processor of claim 22, wherein the next operation register stores an address used for Boolean short-circuiting.

25. The processor of claim 22, wherein the end of OR address register stores the address of an instruction immediately following a conjunct comprising an OR clause.

26. The processor of claim 1, wherein the plurality of registers comprise a plurality of single-bit registers.

27. The processor of claim 26, wherein the plurality of single-bit registers comprise an AND truth state register, an OR truth state register, and an indicator for conjuncts comprising OR clauses.

28. A processing method, comprising:

starting an operation related to a Conjunctive Normal Form Boolean expression, wherein the Boolean expression comprises a conjunct;

evaluating the conjunct; and

selectively short-circuiting a portion of the Boolean expression.

29. The processing method of claim 28, wherein the conjunct is a stand-alone term evaluated as an AND operation.

30. The processing method of claim 28, wherein the conjunct comprises an OR clause.

31. The processing method of claim 30, wherein each of a plurality of terms of the conjunct is evaluated as part of an OR operation.

32. The processing method of claim 31, further comprising setting the value of an OR-bit to 'one' if a predetermined term of the plurality of terms evaluates to true.

33. The processing method of claim 32, further comprising setting the value of the OR-bit to 'zero' if the predetermined term of the plurality of terms does not evaluate to true.

34. The processing method of claim 33, further comprising, in a conjunct comprising an OR clause, OR'ing the result of each OR operation with the current value of an OR register.

35. The processing method of claim 34, further comprising, in the event that the OR register has a value of 'one' and an OR conjunct register is set to 'one', evaluating the conjunct to true and short-circuiting to a next conjunct.

36. The processing method of claim 35, further comprising joining an AND operation and the next conjunct and rolling the value of the OR register up to the AND register by AND'ing the value of the OR register with the value of the AND register.

37. The processing method of claim 36, further comprising, in the event that the OR-bit has a value of 'zero' when the AND operation is processed, changing the AND-bit to a value of 'zero'.

38. The processing method of claim 37, further comprising setting the final value of the Boolean expression to false, if the AND-bit has a value of 'zero', and short circuiting the remainder of the Boolean expression.

39. A device polling unit for finding new devices, assigning addresses to those devices, polling those devices for their current states, and updating a random-access memory with those states, the device polling unit comprising:

a maximum device address electrically-erasable programmable read-only memory, wherein the electrically-erasable programmable read-only memory is operable for storing the highest address of all known devices on a system, wherein the electrically-erasable programmable read-only memory comprises an increment line that increments

its value by one whenever it is asserted and a plurality of output lines that continuously output its value;

an *n*-bit incrementing register, wherein the *n*-bit incrementing register is operable for holding an *n*-bit number representing a current address of a device being polled, wherein the *n*-bit incrementing register comprises a reset line that sets the register to 'zero' whenever it is asserted, and wherein the *n*-bit incrementing register further comprises an increment line and a plurality of output lines that continuously output its value to an AND unit and a current address encoder; and

wherein the device polling unit operates in a continuous loop after it is started.

40. The device polling unit of claim 39, wherein, if a new device is found and a new device found line is asserted, the device polling unit assigns a system address to the new device via direct parallel communication or serially via a new address encoder.

41. The device polling unit of claim 39, wherein, if a new device is not found, the *n*-bit incrementing register is incremented, the device polling unit polls the device corresponding to an address in the register, and the device polling unit copies the device's current state into the random-access memory.

42. A device interface unit for listening for new device seek, new address, state enable, and control line assertions and determining whether or not there is work to do as a result of such assertions, the device interface unit comprising:

a new device electrically-erasable programmable read-only memory, wherein the new device electrically-erasable programmable read-only memory comprises an *n*-bit store that is initially set to 'one', and wherein, when a new device seek line is asserted, the *n*-bit store asserts a new device found line.

43. The device interface unit of claim 42, further comprising an address decoder, wherein, if the *n*-bit store is set, it allows an address passed on a new address line to be placed in an n-bit address electrically-erasable programmable read-only memory and the *n*-bit store to be cleared.

44. The device interface unit of claim 43, further comprising a control word decoder, wherein the control word decoder is operable for reading serial bits off of a control line, and wherein, if an address matches the address in the n-bit address electrically-erasable programmable read-only memory, a plurality of control bits output to a device controller to change its state.

45. The device interface unit of claim 44, further comprising an address and state encoder, wherein the address and state encoder is operable for reading bits in parallel that represent the address and state of the device and serially outputs the bits to a receiver.